One front door for AI: deploying governed agents across Grab

Kevin Bowling, Head of AI Foundry, Grab · Case study · Oct 2026

SUMMARY This case study covers how I led Grab from scattered AI pilots and shadow tools to a single enterprise agentic platform used by 18,000+ employees, and what I learned about getting enterprises to production. I set the strategy and made the key design calls. My team of about 40 built and runs the platform, one governed gateway for people and agents, with each user's own access and approvals. Business units are accountable for their results. Daily users of the platform grew from about 2,000 in May to 8,600 in September 2026, more than 1,000 production workflows run across Legal, Finance, Marketing, Recruiting and other teams, and the work was named an inaugural winner of Forrester's 2026 Data & AI Impact Award (APAC).

Daily users	Growth	Production workflows	Internal mini apps	Apps on live systems
8,600	**4x+**	**1,000+**	**3,000+**	**40%**
on the platform	daily users, May to Sep	in daily use	built by teams	of internal mini apps

Table 1. Adoption and impact, September 2026.

1 Context and why now

As Head of AI Foundry at Grab, I led the discovery in October–November 2025: we mapped the shadow AI, pilots and governance gap. In the same phase we evaluated commercially available AI platforms. We set the design in December–January and started building.

What we found was familiar: pilots scattered across teams, context copied by hand between tools, and one-off bots that each handled permissions in their own way. Where sanctioned tools fell short, people used unsanctioned ones and built apps nobody owned after launch. Nobody could say with confidence which agent did what, or on whose behalf. For a US-listed company in regulated markets, that is a real problem.

We treated Grab's business units as customers, and none of them were captive. If the platform didn't work for them, they would go back to the shadow tools. Leadership asked for one safe, default way to use AI on real work, measured by real usage.

> PLAYBOOK, STEP 1 · DISCOVER In a new enterprise this takes weeks, with the business sponsor and the IT and security leads, mapping pilots, shadow AI and governance gaps against the platform being rolled out.

2 The approach: one governed front door

Four questions were open:

- **Could we just buy it?** Off-the-shelf agents mostly connect to SaaS, not our bought and built systems and private data lakes, and what people build usually stays personal, without the shared flows, agents built by teams and reuse we needed.
- **Which model?** The best model changed every quarter.
- **Who builds the agents?** A central team can't build them all.
- **How do we govern it without slowing adoption or innovation?**

In December 2025 I set the direction: **a single gateway for all AI work**, used by people and agents, acting on someone's behalf or, next, as themselves, with three layers (context preparation, orchestration and governed execution) and access control and audit throughout (Figure 1). My build/buy line: buy frontier models and have my team build only what's specific to Grab. We ruled out an in-house foundation model, hard-wiring one model, and shared service accounts (hard to audit once agents act).

My team built and runs the platform: the gateway, the cross-enterprise connectors and the main app people use. Governance was designed in from the start, so the safe, supportable way to use AI is also the easiest one. Business teams own their processes and results, and my team provides the tooling and the guardrails around it. On sequencing, we started with **governed personal productivity**, so employees could see what was possible, then used that demand to move to **team workflows** and later **organisation-level productivity**.

> PLAYBOOK, STEP 2 · FIRST WORKFLOW Start with individual use on the platform, then build one high-value workflow as a shared agent, measured before and after in terms the CFO already tracks.

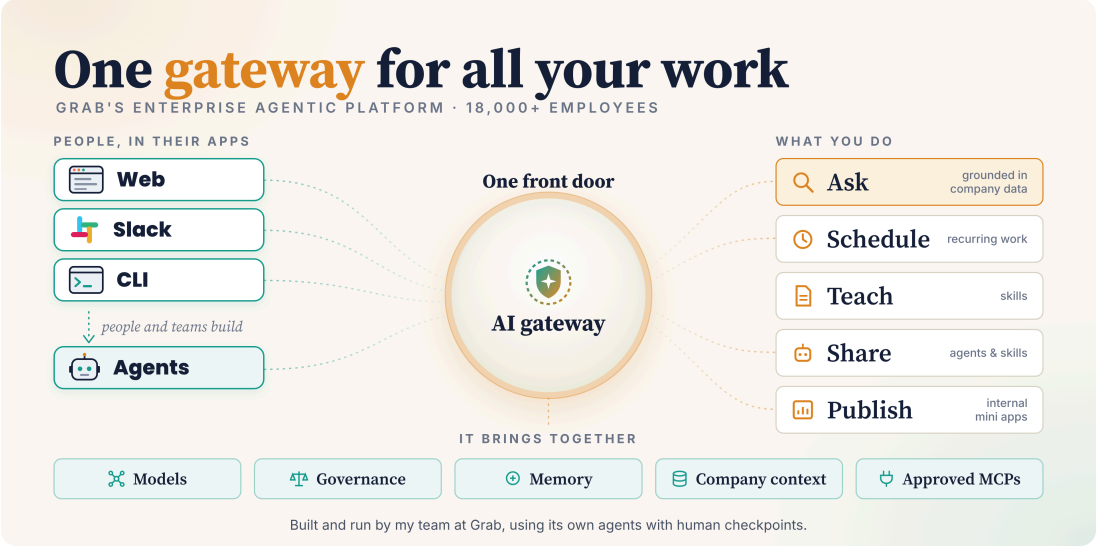


Figure 1. The gateway, used by people from the web, Slack and the command line, and by the agents they build. It applies each user's own access, approval and budget, logs every action, and brings together models, governance, memory, company context and approved MCP connectors.

3 Key design decisions

I made five decisions that shaped the platform.

3.1 **One sign-in for budget, connectors and access.** Each person's token budget, connectors and access are wired to them through one simple sign-in. It was the first time Grab gave every employee one sanctioned, governed way into AI, whether from the app, Slack or the command line, under their own identity and limits. Before that it was a patchwork of separate tools, some of them outside our acceptable-use policy. *Next step:* per-user access works well for personal productivity, but team workflows and "superflows" across business units can't depend on one person's session. My teams are now exploring SPIFFE to give ephemeral agent workloads their own short-lived, verifiable identities, scoped and audited per workflow.

3.2 **Approval scaled to risk, with a full audit trail.** We chose approval by risk over full autonomy or approving every step. In what my team built, reads run without approval. Changes get an approval card, granted once or for a fixed period, and requests expire if nobody answers. High-risk actions need typed confirmation, and external content is treated as data, never as instructions. Every action is logged against the real tool and the real user. *What we'd change:* finer scopes, by recipient or environment, for the riskiest tools.

3.3 **Layered context rather than one big index.** Context comes in layers: search to discover, catalogues to find the authoritative source, structured lookups for system-of-record data, and skills written around Grab's own processes, versioned and loaded only when relevant. *What we'd change:* evaluate skills and check their freshness from day one, because stale skills fail quietly.

3.4 **One agent runtime on an open standard, with model optionality.** We built it so we aren't tied to one model. My team builds the cross-enterprise connectors, and other teams build and certify their own with our harness; all of them, plus internal services, plug in through MCP after review. Today the LLM gateway lets us use frontier and open-weight models; routing each task by difficulty and cost is what we're exploring next. *The trade-off:* an open standard lets any team plug in, but we depend on systems we don't control, so the gateway sets its own timeouts.

3.5 **Internal apps on a paved road.** A small tool used to wait **months** for engineering, or a team bought another SaaS app for one feature. Now there are two ways onto the road my team built: build a mini app with **a few prompts** and deploy it in **seconds**, with login and observability included, or bring your own coding tool, with a small SDK and publish-time checks keeping it on the road. Either way, apps reuse the identity and connector layer, so cybersecurity review is faster and more automated. Teams have built 3,000+ internal mini apps.

> **PLAYBOOK, STEP 3 · GOVERN EARLY** Set up the platform properly before scaling: SSO, SCIM and role-based access, approved apps, and an audit trail feeding the security team's tooling from day one. Connect systems once through MCP, and hand the agents and skills to the business teams to maintain.

4 Rolling it out across business units

We started where one workflow could show results quickly, then ran several units in parallel through the same steps: find the one or two workflows that matter, redesign the process before building, measure before and after with the sponsor, expand from personal to team to organisation use, and train the unit to build and maintain its own agents and workflows.

That was **first-pass review of legal contracts**, starting with standard ones like NDAs, where it went from **30–60 minutes** to **1–2 minutes**. Flagged contracts go to a lawyer. Legal owns the playbook and the sign-off, and my team owns the platform. My teams then deployed team and organisation workflows: order-to-cash in Finance and Procurement, "no-drama" outage communications in Marketing, and now Recruiting.

Grab's ExCo set the direction: every corporate team would define its own vision for working with AI, and our platform would be the enabler. I designed and built a formal RFC process for platform features and key design changes. The hardest call came with pushback from teams who wanted the latest desktop agent tool connected directly to enterprise systems. We put the trade-off to leadership: speed for teams that wanted the newest tool on one side, and governance, observability and reuse on the other. Leadership backed the platform, and where we can, my team now routes those tools through the gateway.

My team also uses the platform to build the platform. Our harness lets coding agents, Codex among them, work on their own from ticket to merge request (MR), through the same gateway, approvals and audit trail as everyone else. They take the report, investigate, write tests that fail first, build the fix, get an independent agent review, and check that the change answers the original ask. Senior developers then review and approve each MR before it goes live, and a daily sweep files anything missed. That lets a team of about 40 ship faster while keeping the quality bar where it was.

> **PLAYBOOK, STEP 4 · EXPAND AND ENABLE** Use the same five steps with an executive sponsor, and bring Security in from week one. Build reusable pieces once and share them across units, such as MCP connector templates for common systems, eval harnesses, adoption-measurement kits and reference agent patterns, and feed product gaps back to the platform vendors.

5 Results and lessons

Table 1 shows adoption, which is still growing, and the contract review in section 4 shows what one workflow can do. The share of apps on live systems matters most, because those apps do real work. The goal is for each function to do more, roughly **doubling capacity at current headcount**. Hours saved is a leading indicator; the measure we hold ourselves to is **output per FTE** in each function. Three lessons so far:

1. **Trust comes from evals, not capability.** People trusted agents they could check: traces collected and labelled, the good ones turned into datasets, and evals run against them to catch regressions across runs. We're now building that loop into the platform so everyone who builds agents gets it.
2. **Context and the ability to act mattered more than model upgrades.** Users came for connectors built around how Grab works, and for agents that could act across Grab's own systems.
3. **Real examples did more than roadmaps.** A before/after told by the business owner moved other units faster than any platform presentation.

> **PLAYBOOK, STEP 5 · MEASURE AND SCALE** Agree an output-per-FTE measure with the CFO, and track daily use in the platform's usage analytics. To scale the team, organise engineers by business area, hire for engineering depth and judgement, and run a weekly review of what the team sees in the field.

Why this carries over. I've been the enterprise customer at this scale, and I've led a customer-facing team as Workato's APAC Head of Enterprise Architecture (2023–25). Many enterprises are where Grab was in late 2025, and this is how I'd approach the same work elsewhere.